How to test indoor positioning

Olli-Pekka Puolitaival 2.6.2016
IndoorAtlas

- Indoor positioning service provider
- Hybrid cloud based software solution
- Development sites in Oulu, Helsinki and Tampere

Numbers:
- 50+ employees
- 15,000 Crowdsourced buildings
- 1-2 meter positioning accuracy
- 16 PhD’s
- 100+ countries
Production infrastructure

- Multi-region cloud services deployed on Microsoft Azure and Amazon Web Services
- Scalable software architecture built on Docker, Mesos, Akka, Cassandra, and much more
- Using best-in-class tools for management and support
Our Platform Workflow

1. Create location
2. Collect data
3. Manage & publish data
4. Build app

Start rockin' it!
How positioning works?

Application

IndoorAtlas SDK

Client

IndoorAtlas positioning service

Cloud

observations

Location information
Testing challenges
Devices are different

Image: https://www.android.com/
Buildings are different
Use cases are different
Applications use our SDK different ways

```
1 - (void)authenticateAndRequestLocation
2
3 {
4    // Create IALocationManager and point delegate to receiver
5    self.manager = [[IALocationManager new] autorelease];
6    self.manager.delegate = self;
7
8    // Set IndoorAtlas API key and secret
9    [self.manager setApiKey:kAPIKey andSecret:kAPISecret];
10
11    // Set initial location
12    IALocation *location = [[IALocation locationWithFloorPlanId:kFloorplanId] autorelease];
13    self.manager.location = location;
14
15    // Request location updates
16    [self.manager startUpdatingLocation];
17 }
```
How we test that all?
1. Functional testing

- Normal software testing methods
- Just to be sure that nothing trivial went broken
- We use all relevant testing methods as part of our integration flow

- Benefits:
  - Tells that basic functionality works

- Limitations:
  - Does not tell how things works in real life

- Tools:
  - Jenkins, unittest frameworks, self made test automation frameworks
2. Monitoring with tests

- Python script that simulates sdk behaviour
- That is run on docker multiple places all the time

- Benefits:
  - Tells that whole positioning service works all the time

- Limitations:
  - Reactive not proactive

- Other tools that we use for monitoring:
  - Datadog, loggly, pingdom, pagerduty, statuspage.io etc.
3. Non-functional performance testing

- Large set of collected data with reference path
- Run algorithms with that data as part of continuous pipeline
- Calculate metrics to tell positioning correctness
- Metrics tells how the change affected to the system performance

- **Benefits:**
  - Show positioning performance correctness

- **Limitations:**
  - No proof that data set represents well real world cases
  - Does not tell how things works in real life

- **Tools:**
  - Jenkins, spark, keen.io
4. Real life data analysis

- **Benefits:**
  - See if things goes really wrong
  - How real use cases looks like
  - How applications are using our sdk
  - How is the internet connection
  - Can give light of our expectation correctness

- **Limitations:**
  - We don’t know where the device really was
  - We don’t know about user at all

- **Tools:**
  - Keen.io, intercom, self made tools
Key learnings

1. Test your expectations
2. Keep learning from real production data
3. Be more data driven than intuition driven
4. Keep clear what is most important just now!
Reading

Data Driven Quality:
https://blogs.msdn.microsoft.com/steverowe/2014/06/16/data-driven-quality/